72. (new) The system of claim 67, further comprising means for storing a program and a remote program revision status in the memory of the remote computer, the remote program revision status indicating the revision level of the program stored in the memory of the remote computer, means for maintaining the latest revisions of the program and a main program revision status in the memory of the main computer, the main program revision status indicating the revision level of the program stored in the memory of the main computer, means for transmitting the remote program revision status from the remote computer to the main computer, means for comparing the remote program revision status to the main program revision status, means for determining updated portions of the program stored in the main computer that are different from the program stored in the remote computer, means for transmitting the updated portions from the main computer to the remote computer, and means for replacing portions of the program stored in the memory of the remote computer with the updated portions received from the main computer.

REMARKS

New Claims 33 to 72 have been copied from and correspond, respectively, to Claims 1-40 of U.S. Patent No. 5,528,490 to Hill ("the Hill '490 patent"). The instant application is a continuation of Application Serial No. 08/740,043 and was filed specifically to copy claims from, and provoke an interference with, the Hill '490 patent. A Filing Under 37 C.F.R. §1.60 and Request For Declaration Of An Interference Under 37 C.F.R. §1.607 are submitted contemporaneously herewith.

Support In Applicants' Disclosure For Newly Presented Claims 33 to 72

As shown in detail by the following chart, new Claims 33 to 72 are supported by the specification and drawings of the above-identified application.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
33. A method for generating information related to a product, the method comprising the steps of:	The Filepp et al. method provides for, inter alia, generating information related to products using programs and other information (called "objects") stored in a remote computer and/or retrieved from a main computer. Filepp et al. disclose that a user, through an RS, can obtain information and perform transactions regarding a wide variety of products and services. P. 10, lines 33 - p. 11, line 20.
storing and maintaining variable data and constant data related to at least one product and a main revision status in a memory of a main computer, the main revision status indicating the revision level of the constant data stored in the main computer;	Filepp et al. disclose that all available forms of data are stored at a main computer (the main computer includes a file server and concentrator which, together, are called the network delivery system) in the form of an interactive network for maintaining application databases and delivering requested parts of the databases on demand to the plurality of remote computer reception systems ("RSs"). P. 7, lines 15-35. Filepp et al. disclose, at p. 137, line 6- p. 138, line 26 that objects can have different storage candidacy values which dictate whether and for how long objects (program instructions and/or data) are stored at the RS. Filepp et al's first or second candidacy values correspond to "variable data" and actually provide for at least two different levels of variable data: A first candidacy value is applied where the object is very sensitive to time; e.g., news items, volatile pricing information such as might apply to stock quotes, etc. In accordance with this first value, the object will not be permitted to be stored on RS 400, and RS 400 will have to request such objects from delivery system 20 each time it is accessed, thus, assuring currency. A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of apples in a grocery shopping application. Here, while the price might change from day to day, it is unlikely to change during a session. Accordingly the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	P. 137, lines 8-19. Filepp et al's third or fifth candidacy values correspond to "constant data": [W]here the object concerns information sufficiently stable to be maintained between sessions, a third storage candidacy value is set to permit the object to be stored at RS 400 between sessions, on condition that the object will be version check[ed] the first time it is accessed in a subsequent session. P. 137, lines 20-25. Where the object is of a type required to be stored at RS 400, as for example, objects needed to support standard screens, it is coded for storage between sessions. . . However, where such objects are likely to change in the future they may be required to be version checked the first time they are accessed in a session and thus [are] given a fifth storage candidacy value. P. 138, lines 1-7. Variable data thus does not persist on the remote computer beyond, at most, a particular user session; it is retrieved from the network delivery system at which it is stored. Constant data is stored locally but is version checked when it is accessed. Thus, the most current constant data is always stored on the network or main computer. Filepp et al's objects include, inter alia, program instructions (i.e., portions of a program) and data. P. 8, lines 25-28; P. 9, lines 29-30. The revision indicia of a stored program is maintained with the object containing the program. Objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. P. 135, lines 22-28. In preferred form, the version id define two fields, a first field to identify the object version and a second field to identify the object storage candidacy. Since the object's version id is part of the object as noted, the version for the object is thus stored wherever the object is stored. Thus, the latest version level of the object will be at the network delivery system.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
storing constant data related to the at least one product and a remote revision status in a memory of a remote computer, the constant data being a subset of information data related to the at least one product, the remote revision status indicating the revision level of the constant data stored in the remote computer;	RAM and disk cached objects are retained at most for the duration of user sessions (and thus are "variable data"), while objects stored in the stage file are retained between sessions (and thus are "constant data"). The storage control field, located in the header portion of an object, described more fully hereafter as the object "storage candidacy", indicates whether the object is stageable, cacheable or trashable: Stageable objects [i.e., constant data] must not be subject to frequent change or update. They are retained between user sessions on the system. . . Cacheable objects [i.e., variable data] can be retained during the current user session, but cannot be retained between sessions. These objects usually have a moderate update frequency. . . Trashable objects [also representing variable data] can be retained only while the user is in the context of the partitioned application in which the object was requested. Trashable objects usually have a very high update frequency and must not be retained to ensure that the user has access to the most current data. . . Specifically, to effect object storage management, objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. . . P. 134, line 2 - p. 135, line 25. An object storage facility provided in the RS software manages objects remotely stored in a local store including a cache (segmented between available RAM and a fixed size disk file) and stage (fixed size disk file). P. 133, lines 30 - p. 134, line 28.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	Since a data object's version id is part of the object, the version for the object is stored wherever the object is stored. For constant data stored at the remote computer, its revision status is also stored there. Filepp et al. describe storing constant information at a remote computer along with a version id that is checked against the version id of corresponding information stored at the network delivery system: [W]here the object concerns information sufficiently stable to be maintained between sessions, a third storage candidacy value is set to permit the object to be stored at RS 400 between sessions, on condition that the object will be version check[ed] the first time it is accessed in a subsequent session. P. 137, lines 20-25. Where the object is of a type required to be stored at RS 400, as for example, objects needed to support standard screens, it is coded for storage between sessions. . . However, where such objects are likely to change in the future they may be required to be version checked the first time they are accessed in a session and thus [are] given a fifth storage candidacy value. P. 138, lines 1-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
transmitting the remote revision status from the remote computer to the main computer;	Filepp et al. disclose that the version of objects available remotely at the RS is compared to the version stored at the main computer and updated at the RS if necessary: When objects are requested from object storage facility 439, only the latest version of the object will be provided to guarantee currency of information to the user. Object storage facility 439 assures currency by requesting version verification from network 10 for those objects which are available locally and by requesting objects which are not locally available from delivery system 20 [i.e., main computer] where currency is maintained. P. 133, lines 7-13. As described above, Filepp et al. describe storing constant information at a remote computer along with a version id that is checked against the version id of corresponding information stored at the network delivery system. P. 137, lines 20-25; P. 138, lines 1-7.
comparing the remote revision status with the main revision status;	Filepp et al. disclose that the local version of an object stored at the RS is checked against the version stored at the main computer: The version value of the object ... provides a parameter that can be checked against predetermined values available from delivery system 20 to determine whether an object stored at RS 400 is sufficiently current to permit its continued use, or whether the object has become stale and needs to be replaced with a current object from delivery system 20. P. 135, line 36 - p. 136, line 5. As described above, Filepp et al. describe storing constant information at a remote computer along with a version id that is checked against the version id of corresponding information stored at the network delivery system. P. 137, lines 20-25; P. 138, lines 1-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
updating constant data stored in the memory of the remote computer with constant data maintained in the memory of the main computer that is different from the constant data stored in the memory of the remote computer;	Filepp et al. disclose updating stale data at the remote RS after version checking: [D]elivery system 20 [i.e., main computer] will advise the reception system 400 either that the version i.d. of the stored object matches the currency value, i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale. P. 139, lines 27-30.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
transmitting variable data related to the at least one product from the main computer to the remote computer; and	Filepp et al. disclose that frequently changing data (i.e., variable data) does not persist at the remote RS beyond, at most, a particular user session and hence must be transmitted as needed from the network: Cacheable objects [i.e., variable data] can be retained during the current user session, but cannot be retained between sessions. These objects usually have a moderate update frequency... Trashable objects [another type of variable data] can be retained only while the user is in the context of the partitioned application in which the object was requested. Trashable objects usually have a very high update frequency and must not be retained to ensure that the user has access to the most current data. P. 134, lines 17-28. A first candidacy value is applied where the object is very sensitive to time; e.g., news items, volatile pricing information such as might apply to stock quotes, etc. In accordance with this first value, the object will not be permitted to be stored on RS 400, and RS 400 will have to request such objects from delivery system 20 [i.e., main computer] each time it is accessed, thus, assuring currency. A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of apples in a grocery shopping application. Here, while the price might change from day to day it is unlikely to change during a session. Accordingly, the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions. Filepp et al., p. 137, lines 8-19.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
integrating constant data related to the at least one product with the variable data related to the at least one product in the remote computer to generate the information data related to the at least one product including both constant data and variable data.	Filepp et al. give the example of a user at the remote computer purchasing an apple through the network. At p. 137, lines 13-19, the price of an apple is described as data transmitted from the network delivery system (i.e., variable data) because it changes so frequently that there is no point in storing it locally. At p. 148, line 26 - p. 153, line 10, the entire procedure by which the user interacts with the remote computer and the network delivery system to purchase apples is detailed. Again, at p. 149, line 36, the price of an apple is obtained from the network delivery system (or main computer) after being selected at the remote computer. The presentation data etc. related to the interactive apple purchase (i.e., constant data) is stored remotely because it does not change frequently. The constant presentation data etc. related to the purchase of apples is clearly shown in Filepp Fig. 3b, with blank spaces for the variable price data which is ultimately transmitted from the network delivery system. Thus, Filepp et al. disclose, inter alia, integrating constant data related to an apple purchase stored at an RS with variable data related to, e.g., the price of an apple obtained from the network delivery system.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
34. The method of claim 33, further comprising the step of selecting a product from the memory of the remote computer for which product information is desired prior to the step of transmitting the remote revision status from the remote computer to the main computer.	Filepp et al. disclose that a user, through an RS, can obtain information and perform transactions regarding a wide variety of products and services. P. 10, lines 33 - p. 11, line 20. Filepp et al. give the example of a user at a remote computer purchasing an apple through the network. At p. 148, line 26 - p. 153, line 10, the entire procedure by which the user interacts with the remote computer and the network delivery system to purchase apples is detailed. Again, at p. 149, line 36, the price of an apple is obtained from the network delivery system (or main computer) after being selected at the remote computer. The presentation data etc. related to the interactive apple purchase (i.e., constant data) is stored remotely because it does not change frequently. The constant presentation data etc. related to the purchase of apples is clearly shown in Filepp Fig. 3b, with blank spaces for the variable price data which is ultimately transmitted from the network delivery system. Filepp et al. further elaborate with regard to the apple purchase example: A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of applies in a grocery shopping application. Here, while the price might change from day to day, it is unlikely to change during a session. Accordingly the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions. P. 137, lines 13-19. Thus, the price of an apple is described as data transmitted from the network delivery system (i.e., variable data) because it changes so frequently that there is no point in storing it at the RS.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
35. The method of claim 34, further comprising the step of automatically connecting the remote computer to the main computer after the selecting step.	Filepp et al. disclose that a remote RS accesses the network through a modem. P. 14, lines 5-7. In the apple purchase example, explained above, the network delivery system (or main computer) is accessed when necessary in response to the user's input regarding the purchase. Automatically connecting the remote computer to the main computer after selecting a product is obvious in view of the Filepp et al. disclosure.
36. The method of claim 35, further comprising the step of automatically disconnecting the remote computer from the main computer after the variable data related to the selected product is transmitted from the main computer to the remote computer.	See claim 35 above. Automatically disconnecting the remote computer is obvious in view of the Filepp et al. disclosure.
37. The method of claim 33, further comprising the step of displaying the information related to the product generated by the remote computer during the integrating step.	For the apple purchase example described above, the display of information related to apples is disclosed. P. 149, lines 36; p. 151, lines 11-14; see also Fig. 3b.
38. The method of claim 33, further comprising the step of printing the information related to the product generated by the remote computer during the integrating step.	Filepp et al. disclose that the remote RS is a standard personal computer with monitor, keyboard and associated elements. P. 8, lines 1-6. A printer is normally attached to personal computers and the information available at an RS is inherently and necessarily printable.
39. The method of claim 33, wherein the constant data stored in the memory of the main computer and the constant data stored in the memory of the remote computer includes both graphics data and textual data.	Filepp et al. disclose objects containing both graphics and textual data. P. 142, lines 17-21. Filepp et al. disclose that display text and graphics necessary to make up addressable partitions constituting the user's display screens are formatted from pre-created objects. P. 17, lines 3-13. These pre-created objects contain constant data.
40. The method of claim 33, further comprising the step of transmitting a request for variable data related to the selected product from the remote computer to the main computer prior to the step of transmitting variable data from the main computer to the remote computer.	See the apple purchase example, discussed above. P. 148, line 26 - p. 153, line 10.
41. The method of claim 33, further comprising the step of transmitting a map from the main computer to the remote computer along with the variable data to permit the remote computer to perform the integrating step.	Filepp et al. disclose transmitting information from the network necessary to process an object requested from the network by the RS. P. 150, lines 23 - p. 151, line 19. See the apple example described above with respect to claim 33.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
42. The method of claim 33, wherein the constant data updating step includes the steps of:	
determining updated portions of the constant data stored in the main computer that are different than the constant data stored in the remote computer;	Filepp et al. disclose checking the remotely stored data against corresponding data available from the network delivery system (or main computer): The version value of the object ... provides a parameter that can be checked against predetermined values available from delivery system 20 [i.e., main computer] to determine whether an object stored at RS 400 is sufficiently current to permit its continued use, or whether the object has become stale and needs to be replaced with a current object from delivery system 20 [i.e., main computer]. P. 135, line 36 - p. 136, line 5.
transmitting the updated portions of the constant data stored in the main computer from the main computer to the remote computer; and replacing portions of the constant data stored on the remote computer with the updated portions of constant data received from the main computer.	Filepp et al. disclose transmitting updated data from the network delivery system to replace stale data stored at the remote computer: [D]elivery system 20 [i.e., main computer] will advise the reception system 400 either that the version i.d. of the stored object matches the currency value, i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale. P. 139, lines 27-30.
43. The method of claim 42, wherein the updating step further includes the step of transmitting a new remote revision status identical to the main revision status from the main computer to the remote computer.	Because the version id is a part of the object header and accordingly the object itself, as described above, the new version indicia is transmitted from the network delivery system to the RS when a new object is transmitted. P. 135, lines 22-25.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
44. The method of claim 33, further comprising the steps of:	
storing a program and a remote program revision status in the memory of the remote computer, the remote program revision status indicating the revision level of the program stored in the memory of the remote computer;	As recited with respect to claim 33, Filepp et al. disclose that objects are stored at the remote RS in accordance with predetermined storage criteria. P. 18, lines 13-19. The revision indicia of a stored program is maintained with the object containing the program. P. 135, lines 22-25. The currency of objects stored at the RS is established by virtue of the object's storage control parameters and a check of the object's version identification prior to use. P. 10, lines 13-19. The terms "version" and "revision" are equivalents in the art. Alan et al. *Computer Desktop Encyclopedia.* Both terms denote the "currency" of the data to which they are respectively applied.
maintaining the latest revisions of the program and a main program revision status in the memory of the main computer, the main program revision status indicating the revision level of the program stored in the memory of the main computer;	As described with respect to claim 33, Filepp et al. disclose a main computer with memory for storing the most current revision indicia of an object. P. 11, line 31 - p. 12, line 4 and P. 13, lines 1-10. Objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. P. 135, lines 22-28. As noted above, objects carry their version id with them in their respective headers and accordingly, wherever the object is stored, so too is its version id. Since the object's version id is part of the object, the latest version level of the object will be at the network delivery system. P. 13, lines 1-10.
transmitting the remote program revision status from the remote computer to the main computer;	As noted with respect to claim 33, Filepp et al. disclose that their system includes transmitting version indicia from a remote RS to the main computer. P. 146, line 27 - p. 148, line 15. As explained at p. 139, lines 23-36, as part of the object request procedure, the RS transmits the object version to ascertain the currency of the object. When a remote RS calls an object for use, the RS first sends a request to the delivery system (i.e., main computer) to verify the currency of the requested object. As part of that request, the RS sends the version and object id for the object.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
comparing the remote program revision status to the main program revision status; and	Filepp et al. disclose comparing version indicia of objects stored at the remote computer and the network delivery system (i.e., main computer). Specifically, during version checking, when an object stored at a remote computer (RS) is initially fetched or accessed during a session, a request to the delivery system (i.e., the main computer) is made to verify object currency by specifying the version id of the object stored at the remote computer. P. 139, lines 23-36. In response, the version id for a referenced object (i.e., the object at the remote RS) is compared to the object version stored at the network delivery system. If the delivery system determines the object version id is current it advises the RS that the object can be used. If the delivery system determines the object is not current, a new object (i.e., the current object) is sent.
updating portions of the program stored in the memory of the remote computer that are different from the program stored and maintained in the memory of the main computer.	Filepp et al. disclose distinguishing between different versions of parts of programs as discussed above. Filepp et al. describe individual object version checking. P. 139, lines 23-36. Filepp et al. describe the formulation of applications, programs and display data with objects. P. 12, lines 7-22. Objects may contain other objects and may also provide reference to other objects by name. P. 9, line 22 - p. 10, line 4. Filepp et al. disclose that program objects are dynamically invoked from other objects, for example, program objects may be called for execution by means of program call segments, "which specify when a program is to be executed (event), what program to execute (program pointer), and how programs should run (parameters)." P. 18, lines 31 - p. 19, line 11. See also p. 13, lines 16-35.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	Filepp et al. disclose the transmission of current versions of programs from the network delivery system (i.e., main computer) to the remote RS, as discussed above. Thus, for example, in response to a request for an object version check, the main computer (i.e., the network delivery system) will advise the remote computer "either that the version id of the stored object matches the currency value; i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale." P. 139, lines 23-36. Filepp et al. disclose replacing an outdated portion of a program stored at the RS with a current version received from the main computer, as discussed above. P. 139, lines 27-30. Because the version id is a part of the object header and accordingly the object itself, as described above, the new version indicia is transmitted from the network delivery system to the RS when a new object is transmitted. P. 135, lines 22-25.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
45. The method of claim 44, wherein the program updating step includes the steps of:	
determining updated portions of the program stored in the main computer that are different from the program stored in the remote computer;	As noted with respect to claim 44, Filepp et al. disclose distinguishing between different versions of parts of programs. As also discussed above, Filepp et al. describes individual object version checking. P. 139, lines 23-36. As also noted with respect to claim 44, Filepp et al. describe the formulation of applications, programs and display data with objects. P. 12, lines 7-22. Objects may contain other objects and may also provide reference to other objects by name. P. 9, lines 22 - p. 10, line 4. Program objects are dynamically invoked from other objects, for example, program objects may be called for execution by means of program call segments, "which specify when a program is to be executed (event), what program to execute (program pointer), and how programs should run (parameters)." P. 18, lines 31 - p. 19, line 11. See also p. 13, lines 16-35, concerning objects being portions of applications; e.g., applications are constructed as groups of objects and distributed on demand to a user's RS. On the point of version checking, again see Filepp et al. p. 139, lines 23-36.
transmitting the updated portions from the main computer to the remote computer; and	Filepp et al. disclose the transmission of current versions of programs from the network delivery system (i.e., main computer) to the remote RS, as discussed above. Thus, for example, in response to a request for an object version check, the main computer (i.e., the network delivery system) will advise the remote computer "either that the version id of the stored object matches the currency value; i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale." P. 139, lines 23-36.
replacing portions of the program stored in the memory of the remote computer with the updated portions received from the main computer.	Filepp et al. disclose replacing an outdated portion of a program stored at the RS with a current version received from the main computer, as discussed above. Where a version checked remotely stored object is found to be stale, the new object delivered by the distribution system will replace the old one. P. 139, lines 27-30.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
46. The method of claim 44, wherein the remote program revision status is transmitted to the main computer each time a communication session is initiated between the remote computer and the main computer.	Filepp et al. disclose that the version of objects (programs or data) available at the remote RS is compared to the version stored at the main computer and updated at the RS if necessary: When objects are requested from object storage facility 439, only the latest version of the object will be provided to guarantee currency of information to the user. Object storage facility 439 assures currency by requesting version verification from network 10 for those objects which are available locally and by requesting objects which are not locally available from delivery system 20 [i.e., main computer] where currency is maintained. P. 133, lines 7-13. As described above, Filepp et al. describe storing constant information at a remote computer along with a version id that is checked against the version id of corresponding information stored at the main computer (network delivery system). P. 137, lines 20-25; P. 138, lines 1-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
47. A method for producing information related to a selected product on a remote computer, the method comprising the steps of:	The Filepp et al. method provides for, inter alia, generating information related to products using program instructions and other information (called "objects") stored in a remote computer and/or retrieved from a main computer.
storing and maintaining variable data and constant data related to a plurality of products in a memory of a main computer;	Filepp et al. disclose a main computer (i.e., file server and concentrator together called a network delivery system) which stores data for delivery to a requesting remote RS, and routes data entered by the user or collected at the RS to the network. P. 11, line 31 - p. 12, line 4. Filepp et al. disclose, at p. 136, line 6 - p. 138, line 26, that objects can have different storage candidacy values which dictate whether and for how long objects (program instructions and/or data) are stored at the RS. Filepp et al's first or second candidacy values correspond to "variable data" and actually provide for at least two different degrees of variable data: A first candidacy value is applied where the object is very sensitive to time; e.g., news items, volatile pricing information such as might apply to stock quotes, etc. In accordance with this first value, the object will not be permitted to be stored on RS 400, and RS 400 will have to request such objects from delivery system 20 each time it is accessed, thus, assuring currency. A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of apples in a grocery shopping application. Here, while the price might change from day to day, it is unlikely to change during a session. Accordingly the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	P. 137, lines 8-19. Filepp et al's third or fifth candidacy values corresponds to "constant data": [W]here the object concerns information sufficiently stable to be maintained between sessions, a third storage candidacy value is set to permit the object to be stored at RS 400 between sessions, on condition that the object will be version check[ed] the first time it is accessed in a subsequent session. P. 137, lines 20-25. Where the object is of a type required to be stored at RS 400, as for example, objects needed to support standard screens, it is coded for storage between sessions. . . However, where such objects are likely to change in the future they may be required to be version checked the first time they are accessed in a session and thus [are] given a fifth storage candidacy value. P. 138, lines 1-7. Variable data thus does not persist on the remote computer beyond, at most, a particular user session; it is retrieved from the main computer (network delivery system) at which it is stored. Constant data is stored locally on the RS but is version checked when accessed. Thus, the most current constant data is always stored on the network.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
storing constant data related to a plurality of products in a memory of a remote computer, the constant data being a subset of product information data related to the plurality of products;	RAM and disk cached objects are retained at most for the duration of user sessions (and thus are "variable data"), while objects stored in the stage file are retained between sessions (and thus are "constant data"). The storage control field, located in the header portion of an object, described more fully hereafter as the object "storage candidacy", indicates whether the object is stageable, cacheable or trashable: Stageable objects [i.e., constant data] must not be subject to frequent change or update. They are retained between user sessions on the system. . . Cacheable objects [i.e., variable data] can be retained during the current user session, but cannot be retained between sessions. These objects usually have a moderate update frequency. . . Trashable objects [also representing variable data] can be retained only while the user is in the context of the partitioned application in which the object was requested. Trashable objects usually have a very high update frequency and must not be retained to ensure that the user has access to the most current data. . . Specifically, to effect object storage management, objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. . . P. 134, line 2 - p. 135, line 25.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	As described above, Filepp et al. describe storing constant information at a remote computer along with a version id that is checked against the version id of corresponding information stored at the main computer (network delivery system): [W]here the object concerns information sufficiently stable to be maintained between sessions, a third storage candidacy value is set to permit the object to be stored at RS 400 between sessions, on condition that the object will be version check[ed] the first time it is accessed in a subsequent session. P. 137, lines 20-25. Where the object is of a type required to be stored at RS 400, as for example, objects needed to support standard screens, it is coded for storage between sessions. . . However, where such objects are likely to change in the future they may be required to be version checked the first time they are accessed in a session and thus [are] given a fifth storage candidacy value. P. 138, lines 1-7.
selecting a product from the remote computer memory for which product information is desired;	Filepp et al. disclose that a user, through a remote RS, can obtain information and perform transactions regarding a wide variety of products and services. P. 10, lines 33 - p. 11, line 1. As discussed above, Filepp et al. give the example of a user at a remote computer purchasing an apple through the network. At p. 148, line 26 - p. 153, line 10, the entire procedure by which the user interacts with the remote computer and the network (main computer) to purchase apples is detailed. Again, at p. 149, line 36, the price of an apple is obtained from the network delivery system (or main computer) after being selected at the remote computer. The presentation data etc. related to the interactive apple purchase (i.e., constant data) is stored remotely at the RS because it does not change frequently. The constant presentation data etc. related to the purchase of apples is clearly shown in Filepp Fig. 3b, with blank spaces for the variable price data which is ultimately transmitted from the network delivery system.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	Filepp et al. further elaborate with regard to the apple purchase example: A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of applies in a grocery shopping application. Here, while the price might change from day to day, it is unlikely to change during a session. Accordingly the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions. P. 137, lines 13-19. Thus, the price of an apple is described as data transmitted from the network (corresponding to "variable data") because it changes so frequently that there is no point in storing it at the RS.
comparing constant data in the memory of the remote computer with constant data in the memory of the main computer;	As described above, Filepp et al. disclose that the local version of an object stored at the RS is checked against the version stored at the main computer: The version value of the object ... provides a parameter that can be checked against predetermined values available from delivery system 20 [i.e., the main computer] to determine whether an object stored at RS 400 is sufficiently current to permit its continued use, or whether the object has become stale and needs to be replaced with a current object from delivery system 20 [i.e., main computer]. P. 135, line 36 - p. 136, line 5.
updating constant data in the memory of the remote computer with constant data stored in the memory of the main computer that is different from the constant data stored in the memory of the remote computer;	Filepp et al. disclose updating stale data at the RS after version checking: [D]elivery system 20 [i.e., the main computer] will advise the reception system 400 either that the version i.d. of the stored object matches the currency value, i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale. P. 139, lines 27-30.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
transmitting variable data related to the selected product from the main computer to the remote computer; and	Filepp et al. disclose that frequently changing data (i.e., variable data) does not persist at the remote RS beyond, at most, a particular user session and hence must be transmitted as needed from the main computer (network delivery system): Cacheable objects [i.e., variable data] can be retained during the current user session, but cannot be retained between sessions. These objects usually have a moderate update frequency... Trashable objects [another type of variable data] can be retained only while the user is in the context of the partitioned application in which the object was requested. Trashable objects usually have a very high update frequency and must not be retained to ensure that the user has access to the most current data... P. 134, lines 17-28. A first candidacy value is applied where the object is very sensitive to time; e.g., news items, volatile pricing information such as might apply to stock quotes, etc. In accordance with this first value, the object will not be permitted to be stored on RS 400, and RS 400 will have to request such objects from delivery system 20 each time it is accessed, thus, assuring currency. A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of apples in a grocery shopping application. Here, while the price might change from day to day it is unlikely to change during a session. Accordingly, the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions. P. 137, lines 8-19.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
integrating constant data stored in the memory of the remote computer associated with the selected product with the variable data received from the main computer to provide the product information data related to the selected product including both constant and variable data.	As described above at p. 137, lines 13-19, the price of an apple is data transmitted from the network delivery system (i.e., variable data) because it changes so frequently that there is no point in storing it locally. At p. 148, line 26 - p. 153, line 10, the entire procedure by which the user interacts with the remote computer and the network to purchase apples is detailed. Again, at p. 149, line 36, the price of an apple is obtained from the network delivery system (or main computer) after being selected at the remote computer. The presentation data etc. related to the interactive apple purchase (i.e., constant data) is stored remotely because it does not change frequently. The constant presentation data etc. related to the purchase of apples is clearly shown in Filepp Fig. 3b, with blank spaces for the variable price data which is ultimately transmitted from the network delivery system. Thus, Filepp et al. disclose, inter alia, integrating constant data related to an apple purchase stored at an RS with variable data related to, e.g., the price of an apple obtained from the network delivery system.
48. The method of claim 47, further comprising the step of automatically connecting the remote computer to the main computer after the selecting step.	Filepp et al. disclose that a remote RS accesses the network through a modem. P. 14, lines 5-7. In the apple purchase example, explained above, the network delivery system (or main computer) is accessed when necessary in response to the user's input regarding the purchase. Automatically connecting the remote computer to the main computer after selecting a product is obvious in view of the Filepp et al. disclosure.
49. The method of claim 48, further comprising the step of automatically disconnecting the remote computer from the main computer after the variable data related to the selected product is transmitted from the main computer to the remote computer.	See Claim 48 above. Automatically disconnecting the remote computer is obvious in view of the Filepp et al. disclosure.
50. The method of claim 47, further comprising the step of displaying the information related to the product generated by the remote computer during the integrating step.	For the apple purchase example, described above, the display of information related to a product -- namely apples, is disclosed. P. 149, line 36 - p. 150, line 3; p. 151, lines 11-14,: see also Fig. 3b.
51. The method of claim 47, further comprising the step of printing the information related to the product generated by the remote computer during the integrating step.	Filepp et al. disclose that the RS is a standard personal computer with monitor, keyboard and associated elements. P. 8, lines 1-6. A printer is normally attached to personal computers and the information available at an RS is inherently and necessarily printable.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
52. The method of claim 47, wherein the constant data stored in the memory of the main computer and the constant data stored in the memory of the remote computer includes both graphics data and textual data.	Filepp et al. disclose objects containing both graphics and textual data. P. 142, lines 17-21. Filepp et al. disclose that display text and graphics necessary to make up addressable partitions constituting the user's display screens are formatted from pre-created objects. P. 17, lines 3-13. These pre-created objects contain constant data.
53. The method of claim 47, further comprising the steps of:	
storing and maintaining a main revision status in the memory of the main computer, the main revision status indicating the last time the constant data stored in the main computer was revised; and	As described with respect to Claim 33, Filepp et al. disclose a main computer (delivery system) with memory for storing the most current revision indicia of an object. P. 11, line 31 - p. 12, line 4, and p. 13, lines 1-10. Objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. P. 135, lines 22-28. In preferred form, the version id define two fields, a first field to identify the object version and a second field to identify the object storage candidacy. P. 135, lines 25-28. As noted above, objects carry their version id with them in their respective headers and accordingly, wherever the object is stored, so too is its version id. As noted, since the object's version id is part of the object, the latest version level of the object will be at the network delivery system. P. 13, lines 1-10.
storing a remote revision status in the memory of the remote computer, the remote revision status indicating the last time the constant data stored in the remote computer was revised.	As noted, Filepp et al. disclose a remote computer (RS) with memory for storing programs having revision indicia. P. 7, lines 7-14. Users access the network with a RS which is configured as a conventional personal computer enabled with software in conformity with the invention. The RS includes an INTEL based processor, RAM, ROM, and disk memory. P. 8, lines 1-14. Users access the network with their respective RS through a modem. P. 14, line 5-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	Filepp et al. disclose that the RS includes means for selectively storing programs and display data in the form of objects. The objects are stored at the RS in accordance with a predetermined storage criteria. P. 10, lines 13-19. Filepp et al. disclose that "[o]bjects carry application program instructions and/or information for display at [the] monitor screen... of [the] RS." P. 9, lines 29-30. The revision indicia of a stored program is maintained with the object containing the program. P. 135, lines 22-25. The currency of objects stored at the RS is established by virtue of the object's storage control parameters and a check of the object's version identification prior to use. P. 10, lines 13-19. As discussed above, the terms "version" and "revision" are equivalents in the art. Both terms denote the "currency" of the data to which they are respectively applied.
54. The method of claim 53, wherein the step of comparing constant data in the memory of the remote computer with constant data in the memory of the main computer includes the step of comparing the remote revision status with the main revision status maintained in the main computer.	As described above, Filepp et al. disclose that the local version of an object stored at the RS is checked against the version stored at the main computer: The version value of the object ... provides a parameter that can be checked against predetermined values available from delivery system 20 [i.e. main computer] to determine whether an object stored at RS 400 is sufficiently current to permit its continued use, or whether the object has become stale and needs to be replaced with a current object from delivery system 20. P. 135, line 36 - p. 136, line 5.
55. The method of claim 47, further comprising the step of transmitting a request for variable data related to a selected product from the remote computer to the main computer prior to the step of transmitting variable data from the main computer to the remote computer.	See the apple purchase example, discussed above. P. 148, line 26 - p. 153, line 10.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
56. The method of claim 47, further comprising the step of transmitting a map from the main computer to the remote computer along with the variable data to permit the remote computer to perform the integrating step.	Filepp et al. disclose transmitting information from the network necessary to process an object requested from the main computer by the remote RS. P. 150, line 23 - p. 151, line 19. See the apple example described above with respect to Claim 33. Filepp et al. disclose "a format map consisting of a destination/length specification for each field of the data to be transferred." P. 89, lines 27-29.
57. The method of claim 47, wherein the constant data updating step includes the steps of:	
determining updated portions of the constant data stored in the main computer that are different than the constant data stored in the remote computer;	Filepp et al. disclose checking the remotely stored data against corresponding data available from the main computer: The version value of the object ... provides a parameter that can be checked against predetermined values available from delivery system 20 [i.e. main computer] to determine whether an object stored at RS 400 is sufficiently current to permit its continued use, or whether the object has become stale and needs to be replaced with a current object from delivery system 20. P. 135, line 36 - p. 136, line 5.
transmitting the updated portions of the constant data stored in the main computer from the main computer to the remote computer; and replacing portions of the constant data stored on the remote computer with the updated portions of constant data received from the main computer.	Filepp et al. disclose transmitting updated data from the main computer and replacing stale data stored at the remote computer: [D]elivery system 20 [i.e., main computer] will advise the reception system 400 either that the version i.d. of the stored object matches the currency value, i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale. P. 139, lines 27-30.
58. The method of claim 57, wherein the constant data updating step further includes the step of transmitting a new remote revision status identical to the main revision status from the main computer to the remote computer.	Filepp et al. disclose the transmission of new version indicia from the main computer to the remote RS, as discussed above. The version id is a part of the object header and accordingly the object itself. P. 135, lines 22-25.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
59. The method of claim 47, further comprising the steps of:	
storing a program and a remote program revision status in the memory of the remote computer, the remote program revision status indicating the revision level of the program stored in the memory of the remote computer;	As noted, Filepp et al. disclose a remote computer (RS) with memory for storing programs having revision indicia. P. 7, lines 7-14. The RS includes an INTEL based processor, RAM, ROM, and disk memory. P. 8, lines 1-14. Users access the network with their respective RS through a modem. P. 14, lines 5-7. Filepp et al. disclose that the RS includes means for selectively storing programs and display data in the form of objects. The objects are stored at the RS in accordance with a predetermined storage criteria. P. 10, lines 13-19. Filepp et al. disclose that the revision indicia of a stored program is maintained with the object containing the program. P. 135, lines 22-25. The currency of objects stored at the RS is established by virtue of the object's storage control parameters and a check of the object's version identification prior to use. P. 10, lines 13-19.
maintaining the latest revisions of the program and a main program revision status in the memory of the main computer, the main program revision status indicating the revision level of the program stored in the memory of the main computer;	As described with respect to Claim 33, Filepp et al. disclose a main computer with memory for storing the most current revision indicia of an object. P. 11, line 31 - p. 12, line 4 and p. 13, lines 1-10. Objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header, P. 135, lines 22-28. As noted above, objects carry their version id with them in their respective headers and accordingly, wherever the object is stored, so too is its version id. Since the object's version id is part of the object, the latest version level of the object will be at the network delivery system (or main computer). P. 13, lines 1-10.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
transmitting the remote program revision status from the remote computer to the main computer;	As noted with respect to Claim 33, Filepp et al. disclose that their system includes the transmission of version indicia from a remote RS to the network. P. 146, line 27 - p. 148, line 15. As explained at p. 139, lines 23-36, as part of the object request procedure, the RS transmits the object version to ascertain the currency of the object. When a remote RS calls an object for use, the RS first sends a request to the main computer to verify the currency of the requested object. As part of that request, the RS sends the version and object id for the object. P. 139, lines 23-26.
comparing the remote program revision status to the main program revision status; and	As noted, Filepp et al. disclose comparing version indicia of objects stored at the RS and the main computer. Specifically, during version checking, when an object stored at a remote computer (RS) is initially fetched or accessed during a session, a request to the delivery system (i.e., the main computer) is made to verify object currency by specifying the version id of the object stored at the remote computer. P. 139, lines 23-36. In response, the version id for a referenced object (i.e., the object at the remote RS) is compared to the object version stored at the network delivery system [i.e., main computer]. If the network delivery system determines the object version id is current it advises the RS that the object can be used. If the network delivery system determines the object is not current, a new object (i.e., the current object) is sent.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
updating portions of the program stored in the memory of the remote computer that are different from the program stored and maintained in the memory of the main computer.	Filepp et al. disclose distinguishing between different versions of parts of programs, as discussed above. Filepp et al. describe individual object version checking. P. 139, lines 23-36. As discussed above, Filepp et al. describe the formulation of applications, programs and display data with objects. P. 12, lines 7-22. Objects may contain other objects and may also provide reference to other objects by name. P. 9, line 22 - p. 10, line 4. See also p. 13, lines 16-35, concerning objects being portions of applications; e.g., applications are constructed as groups of objects and distributed on demand to a user's RS. Filepp et al. disclose the transmission of current versions of programs from the network delivery system (i.e., main computer) to the RS, as discussed above. P. 139, lines 23-36. Filepp et al. disclose replacing an outdated portion of a program stored at the RS with a current version received from the main computer, as discussed above. Where a version checked remotely stored object is found to be stale, the new object delivered by the distribution system will replace the old one. P. 139, lines 27-30. Because the version id is a part of the object header and accordingly the object itself, as described above, the new version indicia is transmitted from the network delivery system [or main computer] to the RS when a new object is transmitted. P. 135, lines 22-25.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
60. The method of claim 59, wherein the program updating step includes the steps of:	
determining updated portions of the program stored in the main computer that are different from the program stored in the remote computer;	Filepp et al. disclose distinguishing between different versions of parts of programs, as discussed above. Filepp et al. describes individual object version checking. P. 139, lines 23-36. As discussed above, Filepp et al. describe the formulation of applications, programs and display data with objects. P. 12, lines 7-22. Objects may contain other objects and may also provide reference to other objects by name. P. 9, line 22 - p. 10, line 4. See also p. 13, lines 16-35, concerning objects being portions of applications; e.g., applications are constructed as groups of objects and distributed on demand to a user's RS. On the point of version checking, again see Filepp et al. p. 139, lines 23-36.
transmitting the updated portions from the main computer to the remote computer; and	Filepp et al. disclose the transmission of current versions of programs from the network delivery system (i.e., main computer) to the RS, as discussed above. Thus, for example, in response to a request for an object version check, the network delivery system [or main computer] will advise the remote computer "either that the version id of the stored object matches the currency value; i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale." P. 139, lines 23-36.
replacing the portions of the program stored in the memory of the remote computer with the updated portions received from the main computer.	Filepp et al. disclose replacing an outdated portion of a program stored at the RS with a current version received from the main computer, as discussed above. Where a version checked remotely stored object is found to be stale, the new object will replace the old one. P. 139, lines 27-30.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
61. The method of claim 59, wherein the remote program revision status is transmitted to the main computer each time a communication session is initiated between the remote computer and the main computer.	As described above, Filepp et al. disclose that the version of objects (program instructions and/or data) available locally at the RS is compared to the version stored at the main computer and updated at the RS if necessary: When objects are requested from object storage facility 439, only the latest version of the object will be provided to guarantee currency of information to the user. Object storage facility 439 assures currency by requesting version verification from network 10 for those objects which are available locally and by requesting objects which are not locally available from delivery system 20 [or main computer] where currency is maintained. P. 133, lines 7-13. As described above, Filepp et al. describe storing constant information at a remote computer along with a version id that is checked against the version id of corresponding information stored at the main computer. P. 137, lines 20-25; P. 138, lines 1-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
62. An electronic catalog system comprising:	Filepp et al. describe an electronic system for, inter alia, transmitting information between a remote user and a main computer which is used to provide information and complete transactions regarding products.
a main computer including a main memory for storing variable data, constant data and a main revision status related to at least one product, the main revision status indicating the revision level of the constant data stored in the main memory;	Filepp et al. disclose a network delivery system [or main computer] which stores data for delivery to a requesting remote RS, and routes data entered by the user or collected at the RS to the network. P. 11, line 31 - p. 12, line 4. Filepp et al. disclose, at p. 137, line 6 - p. 138, line 26, that objects can have different storage candidacy values which dictate whether and for how long objects (program instructions and/or data) are stored at the RS. Filepp et al's first or second candidacy values correspond to "variable data" and actually provide for at least two different degrees of variable data: A first candidacy value is applied where the object is very sensitive to time; e.g., news items, volatile pricing information such as might apply to stock quotes, etc. In accordance with this first value, the object will not be permitted to be stored on RS 400, and RS 400 will have to request such objects from delivery system 20 each time it is accessed, thus, assuring currency. A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of apples in a grocery shopping application. Here, while the price might change from day to day, it is unlikely to change during a session. Accordingly the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions. P. 137, lines 8-19. Filepp et al's third or fifth candidacy values corresponds to "constant data":

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	[W]here the object concerns information sufficiently stable to be maintained between sessions, a third storage candidacy value is set to permit the object to be stored at RS 400 between sessions, on condition that the object will be version check[ed] the first time it is accessed in a subsequent session. P. 137, lines 20-25. Where the object is of a type required to be stored at RS 400, as for example, objects needed to support standard screens, it is coded for storage between sessions. . . However, where such objects are likely to change in the future they may be required to be version checked the first time they are accessed in a session and thus [are] given a fifth storage candidacy value. P. 138, lines 1-7. Variable data thus does not persist on the remote computer beyond, at most, a particular user session, it is retrieved from the network delivery system at which it is stored. Constant data is stored locally but is version checked when it is accessed. Thus, the most current constant data is always stored on the network. Filepp et al. disclose that all active objects reside on the file server. P. 13, lines 1-10. As noted above, objects carry their version id with them in their respective headers and accordingly, wherever the object is stored. Thus, the revision status of constant data is stored at the main computer with each object. Filepp et al's objects include, inter alia, program instructions (portions of a program) and data. P. 8, lines 25-28; p. 9, lines 29-30. The revision indicia of a stored program is maintained with the object containing the program. According to this storage plan, the version id define two fields, a first field to identify the object version and a second field to identify the object storage candidacy. P. 135, lines 22-28.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
a remote computer including a remote memory for storing constant data and a remote revision status related to the at least one product, the constant data being a subset of information data related to the at least one product, the remote revision status indicating the revision level of the constant data stored in the remote memory;	RAM and disk cached objects are retained at most for the duration of user sessions (and thus are "variable data"), while objects stored in the stage file are retained between sessions (and thus are "constant data"). The storage control field, located in the header portion of an object, described more fully above as the object "storage candidacy", indicates whether the object is stageable, cacheable or trashable: Stageable objects [i.e., constant data] must not be subject to frequent change or update. They are retained between user sessions on the system. . . Cacheable objects [i.e., variable data] can be retained during the current user session, but cannot be retained between sessions. These objects usually have a moderate update frequency. . . Trashable objects [also representing variable data] can be retained only while the user is in the context of the partitioned application in which the object was requested. Trashable objects usually have a very high update frequency and must not be retained to ensure that the user has access to the most current data. . . Specifically, to effect object storage management, objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. . . P. 134, line 2 - p. 135, line 25.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	As above, since a data object's version id is part of the object, the version for the object is stored wherever the object is stored. Since constant data is stored at the remote computer, its revision status is also stored there. Filepp et al. describe storing constant information at a remote computer along with a version id that is checked against the version id of corresponding information stored at the network delivery system: [W]here the object concerns information sufficiently stable to be maintained between sessions, a third storage candidacy value is set to permit the object to be stored at RS 400 between sessions, on condition that the object will be version check[ed] the first time it is accessed in a subsequent session. P. 137, lines 20-25. Where the object is of a type required to be stored at RS 400, as for example, objects needed to support standard screens, it is coded for storage between sessions. . . However, where such objects are likely to change in the future they may be required to be version checked the first time they are accessed in a session and thus [are] given a fifth storage candidacy value. P. 138, lines 1-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
means for transmitting the remote revision status from the remote computer to the main computer;	Filepp et al. disclose that the version of constant data available locally at the RS is compared to the version stored at the main computer and updated at the RS if necessary: When objects are requested from object storage facility 439, only the latest version of the object will be provided to guarantee currency of information to the user. Object storage facility 439 assures currency by requesting version verification from network 10 for those objects which are available locally and by requesting objects which are not locally available from delivery system 20 [or main computer] where currency is maintained. P. 133, lines 7-13. Filepp et al. describe storing constant information at a remote computer along with a version id that is checked against the version id of corresponding information stored at the main computer: [W]here the object concerns information sufficiently stable to be maintained between sessions, a third storage candidacy value is set to permit the object to be stored at RS 400 between sessions, on condition that the object will be version check[ed] the first time it is accessed in a subsequent session. P. 137, lines 20-25. Where the object is of a type required to be stored at RS 400, as for example, objects needed to support standard screens, it is coded for storage between sessions. . . However, where such objects are likely to change in the future they may be required to be version checked the first time they are accessed in a session and thus [are] given a fifth storage candidacy value. P. 138, lines 1-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
means for comparing the remote revision status with the main revision status; means for selecting portions of the constant data stored in the main memory that are different from the constant data stored in the remote memory;	As described above, Filepp et al. disclose that the local version of constant data stored at the RS is checked against the version stored at the main computer and selectively updated: The version value of the object ... provides a parameter that can be checked against predetermined values available from delivery system 20 [i.e., main computer] to determine whether an object stored at RS 400 is sufficiently current to permit its continued use, or whether the object has become stale and needs to be replaced with a current object from delivery system 20. P. 135, line 36 - p. 136, line 5.
means for transmitting updated portions of the constant data stored in the main memory from the main computer to the remote computer; means for replacing portions of the constant data stored in the remote memory with the updated portions of constant data received from the main computer;	Filepp et al. disclose transmitting and updating portions of constant data and replacing stale data stored at the RS after version checking: [D]elivery system 20 [or main computer] will advise the reception system 400 either that the version i.d. of the stored object matches the currency value, i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale. P. 139, lines 27-30.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
means for transmitting variable data related to a selected product stored in the main memory from the main computer to the remote computer; and	Filepp et al. disclose that frequently changing data (i.e., variable data) does not persist at the remote RS beyond, at most, a particular user session and hence must be transmitted as needed from the main computer to the remote RS: Cacheable objects [i.e., variable data] can be retained during the current user session, but cannot be retained between sessions. These objects usually have a moderate update frequency... Trashable objects [another type of variable data] can be retained only while the user is in the context of the partitioned application in which the object was requested. Trashable objects usually have a very high update frequency and must not be retained to ensure that the user has access to the most current data... P. 134, lines 17-28. A first candidacy value is applied where the object is very sensitive to time; e.g., news items, volatile pricing information such as might apply to stock quotes, etc. In accordance with this first value, the object will not be permitted to be stored on RS 400, and RS 400 will have to request such objects from delivery system 20 each time it is accessed, thus, assuring currency. A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of apples in a grocery shopping application. Here, while the price might change from day to day it is unlikely to change during a session. Accordingly, the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions. Filepp et al., p. 137, lines 8-19.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
means for integrating constant data related to the selected product stored in the remote memory with the variable data related to the selected product received from the main computer to generate said information data related to the selected product including both constant data and variable data.	As described above, Filepp et al. give the example of a user at the remote computer purchasing an apple. At p. 137, lines 13-19, the price of an apple is variable data because it changes so frequently that there is no point in storing it locally. At p. 148, line 26 - p. 153, line 10, the entire procedure by which the user interacts with the remote computer and the main computer to purchase apples is detailed. Again, at p. 149, line 36, the price of an apple is obtained from the network delivery system (or main computer) after being selected at the remote computer. The presentation data etc. related to the interactive apple purchase (i.e., constant data) is stored remotely because it does not change frequently. The constant presentation data etc. related to the purchase of apples is clearly shown in Filepp Fig. 3b, with blank spaces for the variable price data. Thus, Filepp et al. disclose, inter alia, integrating constant data related to an apple purchase stored at an RS with variable data related to, e.g., the price of an apple obtained from the network delivery system (i.e., the main computer).
63. The system of claim 62, further comprising means for generating a map at the main computer and means for transmitting the map from the main computer to the remote computer along with the variable data to permit the integrating means to generate information related to the selected product including both constant data and variable data.	Filepp et al. disclose transmitting information necessary to process an object requested by the RS. P. 150, line 23 - p. 151, line 19. See the apple example described above with respect to claim 33. Filepp et al. disclose "a format map consisting of a destination/length specification for each field of the data to be transferred." P. 89, lines 27-29.
64. The system of claim 62, wherein the means for transmitting updated portions of the constant data stored in the main memory from the main computer to the remote computer also transmits an updated remote revision status identical to the main revision status from the main computer to the remote computer.	Because the version id is a part of the object header and accordingly the object itself, as described above, the new version indicia is transmitted from the network delivery system [i.e., the main computer] to the RS when a new object is transmitted. P. 135, lines 22-25.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
65. The system of claim 62, further comprising means for storing a program and a remote program revision status in the memory of the remote computer, the remote program revision status indicating the revision level of the program stored in the memory of the remote computer,	As discussed above, Filepp et al. disclose a remote computer (RS) with memory for storing programs having revision indicia. P. 7, lines 7-14. Users access the network with a RS which is configured as a conventional personal computer enabled with software in conformity with the invention. The RS includes an INTEL based processor, RAM, ROM, and disk memory. P. 8, lines 1-14. Users access the network with their respective RS through a modem. P. 14, line 5-7. The RS includes means for selectively storing programs and display data in the form of objects. The objects are stored at the RS in accordance with a predetermined storage criteria. P. 10, lines 13-19. Filepp et al. disclose that "[o]bjects carry application program instructions and/or information for display at [the] monitor screen... of [the] RS." P. 9, lines 29-30. The revision indicia of a stored program is maintained with the object containing the program. P. 135, lines 22-25. The currency of objects stored at the RS is established by virtue of the object's storage control parameters and a check of the object's version identification prior to use. P. 10, lines 13-19. As discussed above, the terms "version" and "revision" are equivalents in the art. Both terms denote the "currency" of the data to which they are respectively applied.
means for maintaining the latest revisions of the program and a main program revision status in the memory of the main computer, the main program revision status indicating the revision level of the program stored in the memory of the main computer,	As described with respect to Claim 33, Filepp et al. disclose a main computer with memory for storing the most current revision indicia of an object. P. 11, line 31 - p. 12, line 4 and p. 13, lines 1-10. Objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. P. 135, lines 22-28. As noted above, objects carry their version id with them in their respective headers and accordingly, wherever the object is stored, so too is its version id. Since the object's version id is part of the object, the latest version level of the object will be at the network delivery system. P. 13, lines 1-10.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
means for transmitting the remote program revision status from the remote computer to the main computer, means for comparing the remote program revision status to the main program revision status, and means for determining updated portions of the program stored in the main computer that are different from the program stored in the remote computer, means for transmitting the updated portions from the main computer to the remote computer, and means for replacing portions of the program stored in the memory of the remote computer with the updated portions received from the main computer.	As noted with respect to claim 33, Filepp et al. disclose that their system includes the transmission of version indicia from a remote RS to the network. P. 146, line 27 - p. 148, line 15. As explained at p. 139, lines 23-36, as part of the object request procedure, the RS transmits the object version to ascertain the currency of the object. When a remote RS calls an object for use, the RS first sends a request to the delivery system (i.e., main computer) to verify the currency of the requested object. As part of that request, the RS sends the version and object id for the object. P. 139, lines 23-26. During version checking, when an object stored at a remote computer (RS) is initially fetched or accessed during a session, a request to the delivery system (i.e., the main computer) is made to verify object currency by specifying the version id of the object stored at the remote computer. P. 139, lines 23-36. In response, the version id for a referenced object (i.e., the object at the remote RS) is compared by the network delivery system to the object version stored at the network delivery system. If the network delivery system determines the object version id is current it advises the RS that the object can be used. If the network delivery system determines the object is not current, a new object (i.e., the current object) is sent. Filepp et al. disclose the transmission of current versions of programs from the network delivery system (i.e., main computer) to the RS. Thus, for example, in response to a request for an object version check, the network delivery system will advise the remote computer "either that the version id of the stored object matches the currency value; i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale." P. 139, lines 23-36. Filepp et al. disclose replacing an outdated portion of a program stored at the RS with a current version received from the main computer. Where a version checked remotely stored object is found to be stale, the new object delivered by the distribution system will replace the old one. P. 139, lines 27-30.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
66. The system of claim 62, wherein the means for transmitting updated portions of the constant data stored in the main memory from the main computer to the remote computer also transmits an updated remote revision status identical to the main revision status from the main computer to the remote computer.	Because the version id is a part of the object header and accordingly the object itself, as described above, the new version indicia is transmitted from the main computer (network delivery system) to the remote RS when a new object is transmitted. P. 135, lines 22-25.
67. An electronic catalog system comprising:	Filepp et al. describe an electronic system for, inter alia, transmitting information between a remote user and a main computer which is used to provide information and complete transactions regarding products.
a main computer including a main memory for storing variable data and constant data related a plurality of products;	Filepp et al. disclose that their file server and concentrator together constitute a network delivery system which stores data for delivery to a requesting remote RS, and routes data entered by the user or collected at the RS to the network. P. 11, line 31 - p. 12, line 4. Filepp et al. disclose, at p. 137, line 6 - p. 138, line 26, that objects can have different storage candidacy values which dictate whether and for how long objects (program instructions and/or data) are stored at the RS. Filepp et al's first or second candidacy values correspond to "variable data" and actually provide for at least two different degrees of variable data: A first candidacy value is applied where the object is very sensitive to time; e.g., news items, volatile pricing information such as might apply to stock quotes, etc. In accordance with this first value, the object will not be permitted to be stored on RS 400, and RS 400 will have to request such objects from delivery system 20 [i.e., main computer] each time it is accessed, thus, assuring currency. A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of apples in a grocery shopping application. Here, while the price might change from day to day, it is unlikely to change during a session. Accordingly the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	P. 137, lines 8-19. Filepp et al's third or fifth candidacy values corresponds to "constant data": [W]here the object concerns information sufficiently stable to be maintained between sessions, a third storage candidacy value is set to permit the object to be stored at RS 400 between sessions, on condition that the object will be version check[ed] the first time it is accessed in a subsequent session. P. 137, lines 20-25. Where the object is of a type required to be stored at RS 400, as for example, objects needed to support standard screens, it is coded for storage between sessions. . . However, where such objects are likely to change in the future they may be required to be version checked the first time they are accessed in a session and thus [are] given a fifth storage candidacy value. P. 138, lines 1-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
a remote computer including a remote memory for storing constant data related to a plurality of products, the constant data being a subset of product information data related to the plurality of products;	RAM and disk cached objects are retained at most for the duration of user sessions (and thus are "variable data"), while objects stored in the stage file are retained between sessions (and thus are "constant data"). The storage control field, located in the header portion of an object, described more fully hereafter as the object "storage candidacy", indicates whether the object is stageable, cacheable or trashable: Stageable objects [i.e., constant data] must not be subject to frequent change or update. They are retained between user sessions on the system. . . Cacheable objects [i.e., variable data] can be retained during the current user session, but cannot be retained between sessions. These objects usually have a moderate update frequency. . . Trashable objects [also representing variable data] can be retained only while the user is in the context of the partitioned application in which the object was requested. Trashable objects usually have a very high update frequency and must not be retained to ensure that the user has access to the most current data. . . Specifically, to effect object storage management, objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. . . P. 134, line 2 - p. 135, line 25. An object storage facility provided in the RS software manages objects remotely stored in a local store including a cache (segmented between available RAM and a fixed size disk file) and stage (fixed size disk file). P. 133, line 30 - p. 134, line 28. As described above, Filepp et al. describe storing constant information at a remote computer along with a version id that is checked against the version id of corresponding information stored at the network delivery system. P. 137, lines 20-25; P. 138, lines 1-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
means for transmitting a request for variable data related to a selected product from the remote computer to the main computer;	Filepp et al. disclose that frequently changing data (i.e., variable data) does not persist at the remote RS beyond, at most, a particular user session and hence must be transmitted as needed from the network delivery system: Cacheable objects [i.e. variable data] can be retained during the current user session, but cannot be retained between sessions. These objects usually have a moderate update frequency... Trashable objects [another type of variable data] can be retained only while the user is in the context of the partitioned application in which the object was requested. Trashable objects usually have a very high update frequency and must not be retained to ensure that the user has access to the most current data... P. 134, lines 17-28. A first candidacy value is applied where the object is very sensitive to time; e.g., news items, volatile pricing information such as might apply to stock quotes, etc. In accordance with this first value, the object will not be permitted to be stored on RS 400, and RS 400 will have to request such objects from delivery system 20 each time it is accessed, thus, assuring currency. A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of apples in a grocery shopping application. Here, while the price might change from day to day it is unlikely to change during a session. Accordingly, the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions. Filepp et al., p. 137, lines 8-19.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
means for comparing constant data in the remote memory with constant data in the main memory; means for determining which portions of the constant data stored in the main memory are different from the constant data stored in the remote memory;	As described above, Filepp et al. disclose that the local version of constant data stored at the RS is checked against the version stored at the network: The version value of the object ... provides a parameter that can be checked against predetermined values available from delivery system 20 [i.e., main computer] to determine whether an object stored at RS 400 is sufficiently current to permit its continued use, or whether the object has become stale and needs to be replaced with a current object from delivery system 20. P. 135, line 36 - p. 136, line 5.
means for transmitting updated portions of the constant data stored in the main memory from the main computer to the remote computer; means for replacing portions of the constant data stored in the remote memory with the updated portions of constant data received from the main computer;	Filepp et al. disclose transmitting and updating portions of constant data and replacing stale data at the RS after version checking: [D]elivery system 20 [i.e., main computer] will advise the reception system 400 either that the version i.d. of the stored object matches the currency value, i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale. P. 139, lines 27-30.
means for transmitting variable data related to the selected product stored in the main memory from the main computer to the remote computer; and	Filepp et al. disclose that frequently changing data (i.e., variable data) does not persist at the remote RS beyond, at most, a particular user session and hence must be transmitted as needed from the network delivery system: Cacheable objects [i.e. variable data] can be retained during the current user session, but cannot be retained between sessions. These objects usually have a moderate update frequency... Trashable objects [another type of variable data] can be retained only while the user is in the context of the partitioned application in which the object was requested. Trashable objects usually have a very high update frequency and must not be retained to ensure that the user has access to the most current data... P. 134, lines 17-28.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	A first candidacy value is applied where the object is very sensitive to time; e.g., news items, volatile pricing information such as might apply to stock quotes, etc. In accordance with this first value, the object will not be permitted to be stored on RS 400, and RS 400 will have to request such objects from delivery system 20 each time it is accessed, thus, assuring currency. A second value is applied where the object is sensitive to time but less so than the first case; e.g., the price of apples in a grocery shopping application. Here, while the price might change from day to day it is unlikely to change during a session. Accordingly, the object will be permitted to persist in RAM or at the disk cache during a session, but will not be permitted to be maintained at RS 400 between sessions. Filepp et al., p. 137, lines 8-19.
means for integrating constant data related to the selected product stored in the remote memory with the variable data related to the selected product received from the main computer to generate the product information data related to the selected product including both constant data and variable data.	As described above, Filepp et al. give the example of a user at the remote computer purchasing an apple through the network. At p. 137, lines 13-19, the price of an apple is described as data transmitted from the network (i.e., variable data) because it changes so frequently that there is no point in storing it locally. At p. 148, line 26 - p. 153, line 10, the entire procedure by which the user interacts with the remote computer and the network to purchase apples is detailed. Again, at p. 149, line 36, the price of an apple is obtained from the network delivery system (or main computer) after being selected at the remote computer. The presentation data etc. related to the interactive apple purchase (i.e., constant data) is stored remotely because it does not change frequently. The constant presentation data etc. related to the purchase of apples is clearly shown in Filepp Fig. 3b, with blank spaces for the variable price data which is ultimately transmitted from the network. Thus, Filepp et al. disclose, inter alia, integrating constant data related to an apple purchase stored at an RS with variable data related to, e.g., the price of an apple obtained from the network delivery system.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
68. The system of claim 67, further comprising means for automatically connecting the remote computer to the main computer.	Filepp et al. disclose that a remote RS accesses the network through a modem. P. 14, lines 5-7. In the apple purchase example, explained above, the network delivery system (or main computer) is accessed when necessary in response to the user's input regarding the purchase. Automatically connecting the remote computer to the main computer is obvious in view of the Filepp et al. disclosure.
69. The system of claim 68, further comprising means for automatically disconnecting the remote computer from the main computer after the variable data related to the selected product is transmitted from the main computer to the remote computer.	See Claim 68 above. Automatically disconnecting the remote computer is obvious in view of the Filepp et al. disclosure.
70. The system of claim 67, further comprising means for storing and maintaining a main revision status in the memory of the main computer, the main revision status indicating the revision level of the constant data stored in the main computer,	As described with respect to Claim 33, Filepp et al. disclose a main computer with memory for storing the most current revision indicia of an object. P. 11, line 31 - p. 12, line 4 and p. 13, lines 1-10. Objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. P. 135, lines 22-28. As noted above, objects carry their version id with them in their respective headers and accordingly, wherever the object is stored, so too is its version id. Since the object's version id is part of the object, the latest version level of the object will be at the network delivery system. P. 13, lines 1-10.
and means for storing a remote revision status in the memory of the remote computer, the remote revision status indicating the revision level of the constant data stored in the remote computer.	Filepp et al. disclose a remote computer (RS) with memory for storing programs having revision indicia in the form of a network that features a plurality of such remote computers for displaying information and providing transactional services to users. P. 7, lines 7-14. Users access the network with a RS which is configured as a conventional personal computer enabled with software in conformity with the invention. The RS includes an INTEL based processor, RAM, ROM, and disk memory. P. 8, lines 1-14. Users access the network with their respective RS through a modem. P. 14, lines 5-7.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
	Filepp et al. disclose that the RS includes means for selectively storing programs and display data in the form of objects. The objects are stored at the RS in accordance with a predetermined storage criteria. P. 10, lines 13-19. Filepp et al. disclose that "[o]bjects carry application program instructions and/or information for display at [the] monitor screen... of [the] RS." P. 9, lines 29-30.
71. The system of claim 70, wherein the means for comparing constant data in the remote memory with constant data in the main memory compares the remote revision status with the main revision status maintained in the main computer.	The revision indicia of a stored program is maintained with the object containing the program. P. 135, lines 22-25. The currency of objects stored at the RS is established by virtue of the object's storage control parameters and a check of the object's version identification prior to use. P. 10, lines 13-19. As discussed above, the terms "version" and "revision" are equivalents in the art. Both terms denote the "currency" of the data to which they are respectively applied. Filepp et al. disclose that the local version of an object stored at the RS is checked against the version stored at the network: > The version value of the object ... provides a parameter that can be checked against predetermined values available from delivery system 20 to determine whether an object stored at RS 400 is sufficiently current to permit its continued use, or whether the object has become stale and needs to be replaced with a current object from delivery system 20. P. 135, line 36 - p. 136, line 5.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
72. The system of claim 67, further comprising means for storing a program and a remote program revision status in the memory of the remote computer, the remote program revision status indicating the revision level of the program stored in the memory of the remote computer,	As discussed above, Filepp et al. disclose a remote computer (RS) with memory for storing programs having revision indicia. P. 7, lines 7-14. Users access the network with a RS which is configured as a conventional personal computer enabled with software in conformity with the invention. The RS includes an INTEL based processor, RAM, ROM, and disk memory. P. 8, lines 1-14. Users access the network with their respective RS through a modem. P. 14, lines 5-7. The RS includes means for selectively storing programs and display data in the form of objects. The objects are stored at the RS in accordance with a predetermined storage criteria. P. 10, lines 13-19. Filepp et al. disclose that "[o]bjects carry application program instructions and/or information for display at [the] monitor screen... of [the] RS." P. 9, lines 29-30. Filepp et al. disclose that the revision indicia of a stored program is maintained with the object containing the program. P. 135, lines 22-25. The currency of objects stored at the RS is established by virtue of the object's storage control parameters and a check of the object's version identification prior to use. P. 10, lines 13-19. As discussed above, the terms "version" and "revision" are equivalents in the art. Both terms denote the "currency" of the data to which they are respectively applied.
means for maintaining the latest revisions of the program and a main program revision status in the memory of the main computer, the main program revision status indicating the revision level of the program stored in the memory of the main computer,	As described with respect to Claim 33, Filepp et al. disclose a main computer with memory for storing the most current revision indicia of an object. P. 11, line 31 - p. 12, line 4 and p. 13, lines 1-10. Objects are provided with a coded version id made up of the storage control byte and version control bytes identified above as elements of the object header. P. 135, lines 22-28. As noted above, objects carry their version id with them in their respective headers and accordingly, wherever the object is stored, so too is its version id. Since the object's version id is part of the object, the latest version level of the object will be at the network delivery system. P. 13, lines 1-10.

PROPOSED NEW CLAIMS IN THE PRESENT FILEPP ET AL. APPLICATION	SUPPORT FOR PROPOSED NEW CLAIMS IN FILEPP ET AL. DISCLOSURE
means for transmitting the remote program revision status from the remote computer to the main computer, means for comparing the remote program revision status to the main program revision status, means for determining updated portions of the program stored in the main computer that are different from the program stored in the remote computer, means for transmitting the updated portions from the main computer to the remote computer, and means for replacing portions of the program stored in the memory of the remote computer with the updated portions received from the main computer.	As noted with respect to Claim 33, Filepp et al. disclose that their system includes structure for transmitting version indicia from a remote RS to the network. P. 146, line 27 - p. 148, line 15. As explained at p. 139, lines 23-36, as part of the object request procedure, the RS transmits the object version to ascertain the currency of the object. When a remote RS calls an object for use, the RS first sends a request to the delivery system (i.e., main computer) to verify the currency of the requested object. As part of that request, the RS sends the version and object id for the object. P. 139, lines 23-26. During version checking, when an object stored at a remote computer (RS) is initially fetched or accessed during a session, a request to the delivery system (i.e., the main computer) is made to verify object currency by specifying the version id of the object stored at the remote computer. P. 139, lines 23-36. In response, the version id for a referenced object (i.e., the object at the remote RS) is compared by the network delivery system to the object version stored at the network delivery system. If the network delivery system determines the object version id is current it advises the RS that the object can be used. If the network delivery system determines the object is not current, a new object (i.e., the current object) is sent. Filepp et al. disclose the transmission of current versions of programs from the network delivery system, (i.e., main computer) to the RS. Thus, for example, in response to a request for an object version check, the network delivery system will advise the remote computer "either that the version id of the stored object matches the currency value; i.e., the stored object is acceptable, or deliver a current object that will replace the stored object shown to be stale." P. 139, lines 23-36. Filepp et al. disclose replacing an outdated portion of a program stored at the RS with a current version received from the main computer. Where a version checked remotely stored object is found to be stale, the new object delivered by the distribution system will replace the old one. P. 139, lines 27-30.

CONCLUSION

Entry of the present amendment is respectfully requested.

AUTHORIZATION

An Amendment Fee Transmittal and check for the Preliminary Amendment fee is submitted herewith. However, the Commissioner is hereby authorized to charge any additional fees which may be required for this Preliminary Amendment, or credit any overpayment to Deposit Account No. 13-4500, Order No. 1963-4727. A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

Respectfully submitted,

Israel Blum

September 18, 1997

Israel Blum
Registration No. 26,710

MORGAN & FINNEGAN, L.L.P.
345 Park Avenue
New York, New York 10154
(212) 758-4800
(212) 751-6849 (FAX)